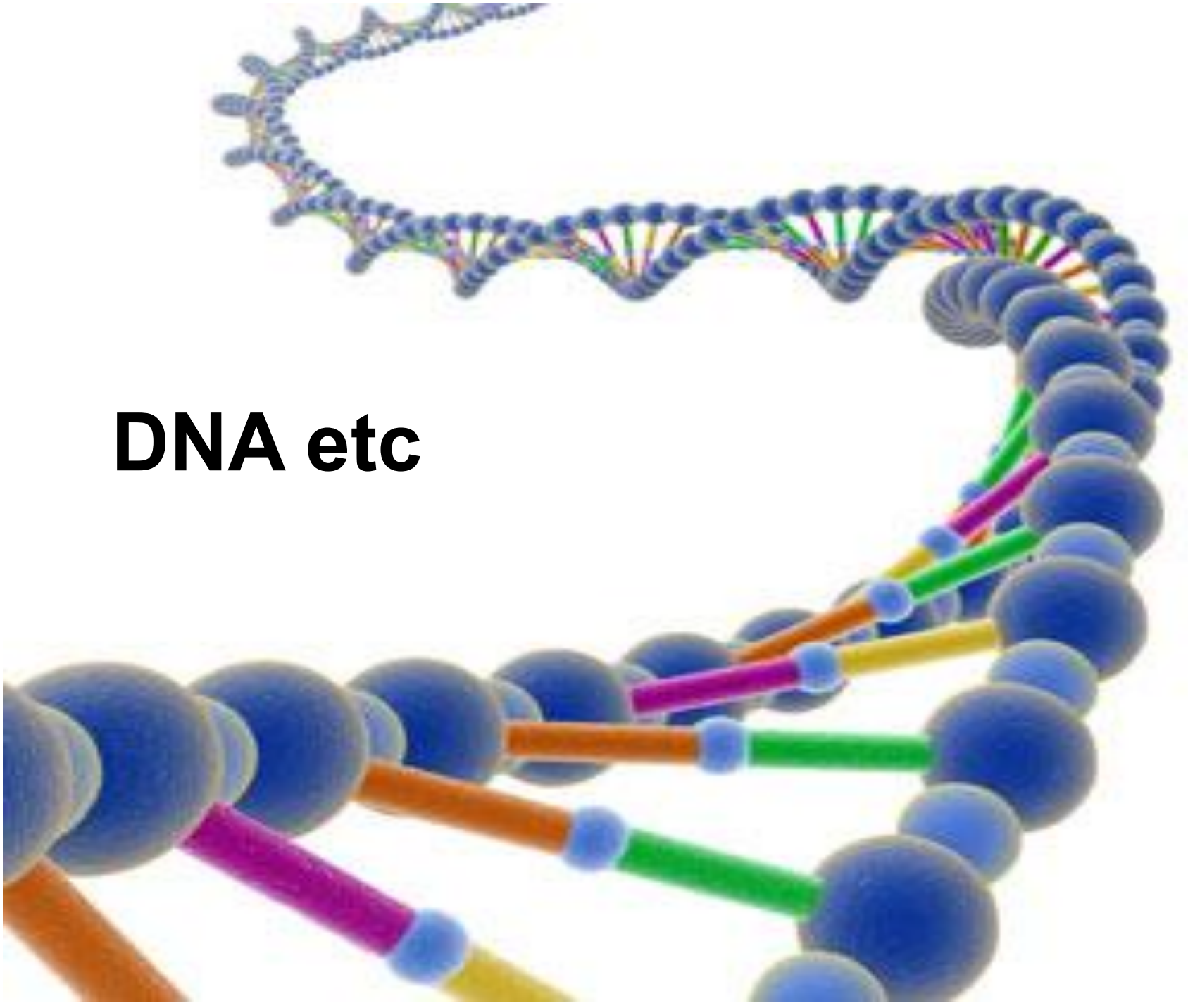
DNA etc



App design rectification

Was:

- $.onDomReady
- MVC modules
- global variables - jQuery, gettext, underscore

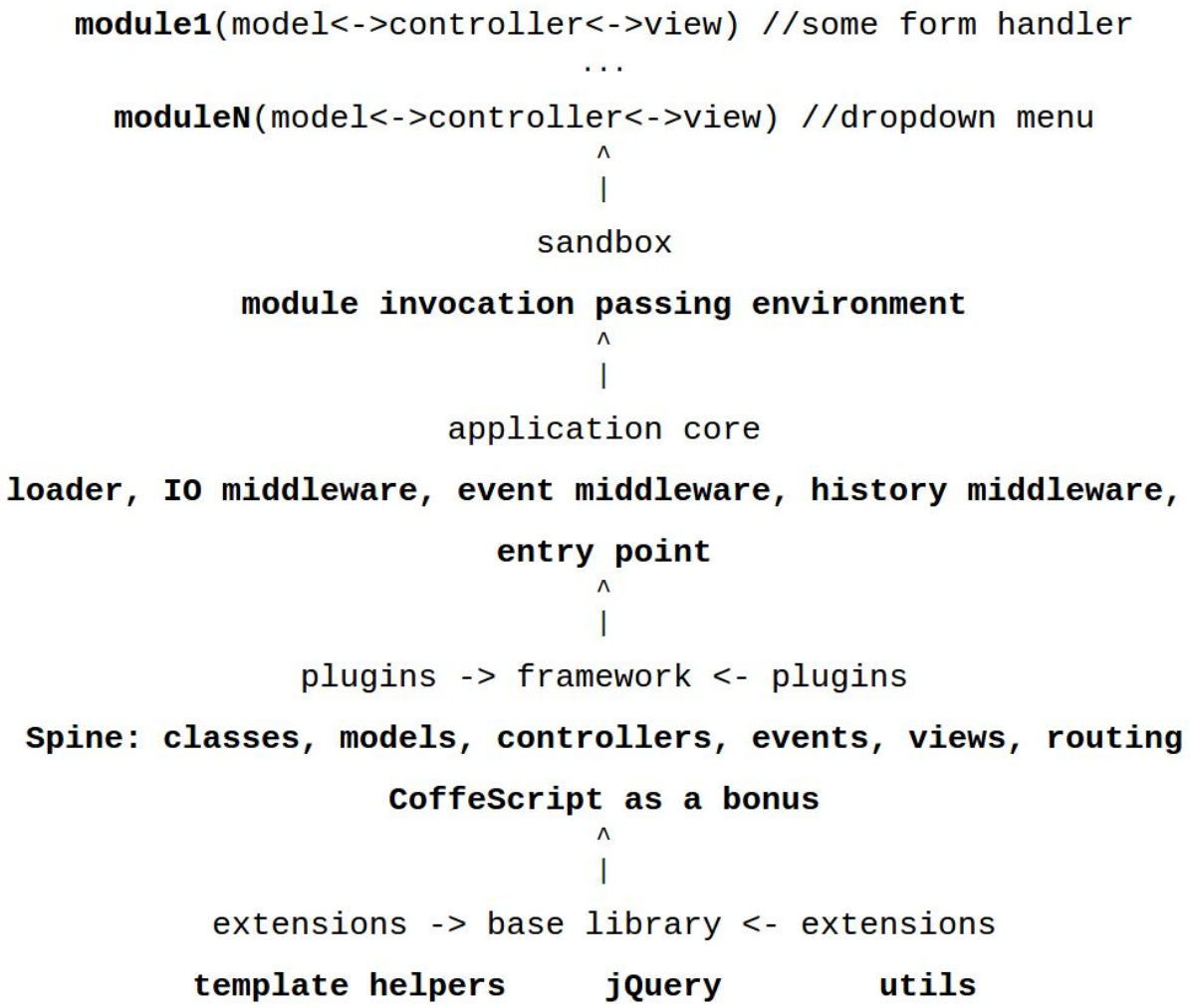
Now:

- bootstrapper
- MVC modules, UI components, services
- abstractions, dispatchable concrete implementations

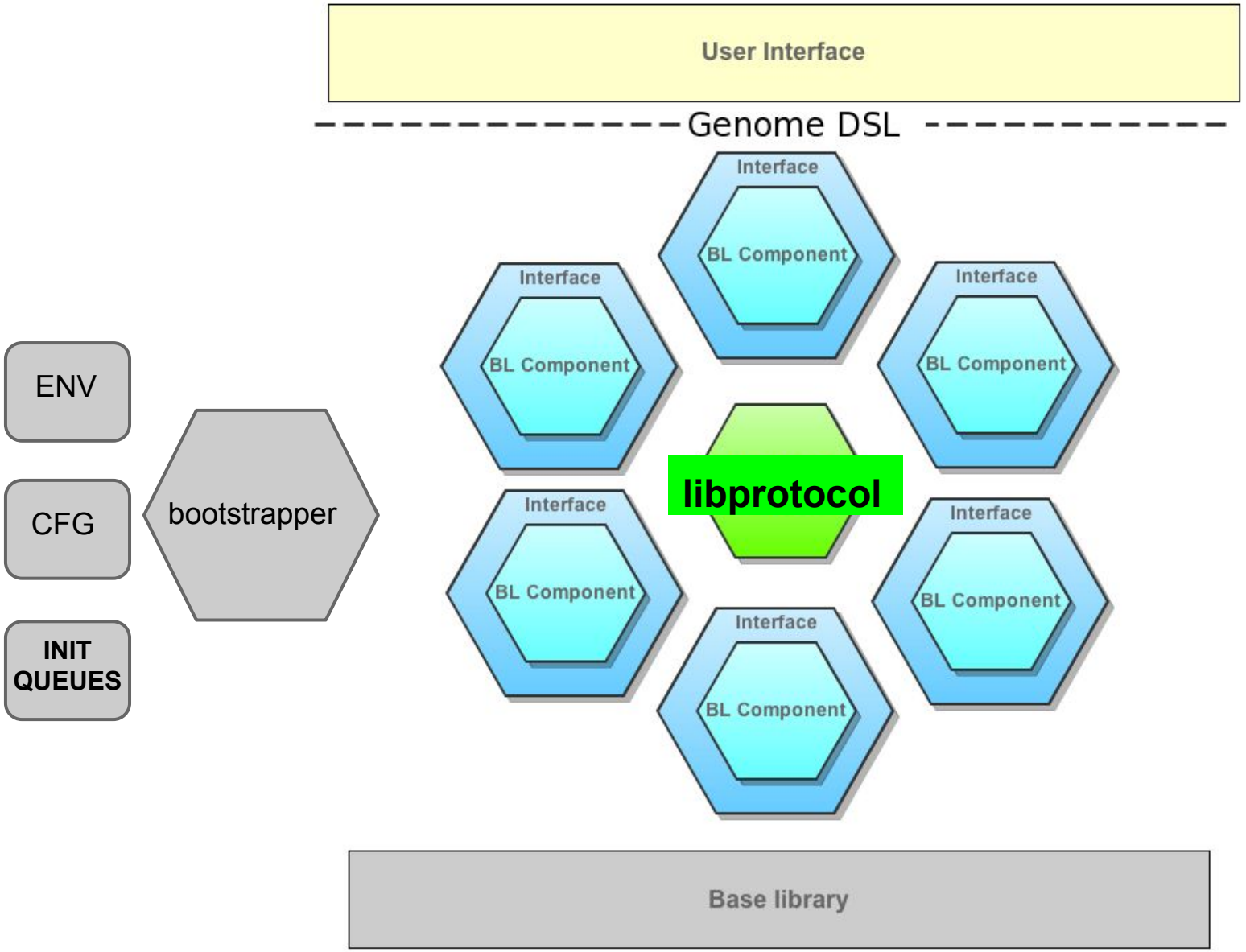
App shortcomings

- css-selectors leaking into code
- DOM accidental complexity and lack of DRY
- impossible to reason which code is bound to which UI structure from UI mrkup
- accidental complexity in manual polymorphism handling
- lack of DRY due to insufficent abstractions
- global variables - jQuery, gettext, underscore

Proposed Prom.ua javascript architecture



DNA


User Interface
Genome DSL
Interface
BL Component
Interface
BL Component
Interface
BL Component
Interface
BL Component
Interface
BL Component
Interface
BL Component
libprotocol
ENV
CFG
INIT
QUEUES
bootstrapper
Base library

Bootstrapper

- provides ENV
- provides app config
- runs init queues
  - init-queue
  - document-ready-queue
  - document-loaded-queue

libprotocol

- interface-based multiple dispatch polymorphism
- metadata

```
Countable = [
    ['count',  [], {async: true, concerns: {before: [do_a, do_b]}}]
]

extend_with Countable, [Array, Object, String,
                                ... also ItemsList, Menu etc]

)
```

DNA

- declarative binding of UI elements with logic
- automatic composition of sync and async methods, error handling, state propagation, AOP, stack-oriented programming support etc
- reactive extensions protocol
- mutation observers
- **data-extend**
- **data-subscribe**

Heavy lifting by **libmonad**

DNA: Genome DSL

```
<div id="cell1"
      data-extend-with="IDraggable IPositionReporter IMovable ">

    <span data-subscribe="
            onDragStop@cell1 : getX@cell1 | setContent">
        here be x </span>

    <span data-subscribe="
            onDragStop@cell1 : getY@cell1 | setContent">
       here be y </span>

    <button data-subscribe='click : "Hello_world" | say'>
       Say hello </button>
</div>
```

DNA: Genome DSL

```
<section id="vector_test">
    <input id="val_a" type="text" placeholder="Value a" />
    +
    <input id="val_b" type="text" placeholder="Value b" />
    =
    <span data-subscribe='
        on_change@val_a,
        on_change@val_b :
            "calculating..." | setText,
            [ getValueAsync@val_a getValue@val_b] | add | swap [NaN "?"] | setText
    '>
        ?
    </span>

</section>
```

DNA: Interfaces

```
{dispatch_impl} = require 'libprotocol'
{info, warn, error, debug} = dispatch_impl 'ILogger', 'Toggler'
{data_to_opts, partial, pubsubhub, to_hash} = require 'libprotein'
jQuery = require 'jquery'

extract_opts = partial data_to_opts, 'togglable'

SHOW = 'show'
HIDE = 'hide'

togglr_proto = [
   ['*cons*',  [], {concerns: {before: [extract_opts, pubsubhub]}}]

   ['hide',    []]
   ['show',    []]
   ['toggle',  []]

   ['on_hide', ['f']]
   ['on_show', ['f']]

]
```

```
togglr_impl = (node, opts, {pub, sub}) ->
   $node = jQuery node

   hide: (ev) ->
      $node.hide()
      pub HIDE

   show: (ev) ->
      $node.show()
      pub SHOW

   toggle: (ev) ->
      $node.toggle()
      if ($node.is ':visible') then (pub SHOW) else
(pub HIDE)

   on_show: partial sub, SHOW

   on_hide: partial sub, HIDE


module.exports =
   protocols:
      definitions:
         ITogglable: togglr_proto
      implementations:
         ITogglable: togglr_impl
```

todomvc demo

The End